University of New Mexico
Department of Computer Science

Final Examination

CS 591-04 Randomized Algorithms
Fall, 2004

- This is the take home final. It's due Monday, Dec 13th at 5pm.
- You may use your textbook and class notes in completing this assignment. *You may not use any other resources*. In particular, you may not speak to other students about the final or consult the web or other outside resources.
- Please type or write up your solutions concisely on separate pages. Include both your name and email with your solutions.
- Note that 3-4 pages should suffice for your solutions. Please do not use much more than this!
- Good Luck!!!

1. **The Probabilistic Method**

   Let $G = (V, E)$ be a graph with n vertices and m edges. Show that G contains a bipartite subgraph with at least $m/2$ edges.

2. **Markov Chains**

   Consider the following undirected graphs over n vertices:

   - *Star*: One vertex v is connected to the other $n-1$ vertices. There are no other edges.
   - *Complete Bipartite*: There are $n/2$ vertices on the left side and $n/2$ vertices on the right side. Every vertex on the left side is connected with every vertex on the right side.
   - *Cycle*: The n vertices are connected in a single cycle with n edges.

   Give the cover times to within Θ for each of these graphs. Justify your answers.

3. **Pattern Matching**

   Consider the following problem. You are given an integer m and two binary strings X and Y both of length n where $n > m$. You want to find all substrings of length m which occur in *both* X and Y.

   The naive way to do this is to apply the pattern matching algorithm done in class repeatedly to each of the $n - m + 1$ substrings of X and the string Y. However this algorithm takes $O((n-m)(n+m))$ time.

   Design an efficient Monte Carlo algorithm for this problem. The expected runtime of the algorithm should also be $O(n+m)$ and the space requirements should be $O(n+m)$ machine words. Show that your algorithm is correct with high probability. You may assume the *unit cost RAM model* i.e. machine words are $O(\log n)$ bits and any arithmetic operations involving $O(\log n)$ bit numbers take constant time (see p. 162 of the text).

4. **Cryptocloning**

   Alice has volunteered for a joint project between the NSA and NIH in cryptocloning. This project has created n Alice clones where n is an odd number. Unfortunately, due to cloning error, $(n-1)/2$ of these Alices are evil and $(n+1)/2$ Alices are good. The good Alices want to send a binary message X of m bits to Bob. The evil Alices try to prevent Bob from knowing X. Bob can not tell the good Alices from the evil ones but he does know m.[1] Although each good Alice knows X, the good Alices can not communicate amongst themselves.

   A simple protocol to ensure that Bob receives the correct message X is the following. Each good Alice sends the message X to Bob. Bob then receives a set of messages $\{X_1, X_2, \ldots, X_l\}$ (one from all the good Alices and possibly some others from the evil Alices). Bob accepts only that unique message which he received from at least $(n+1)/2$ Alices. Unfortunately, this protocol requires the good Alices to send $O(nm)$ bits. In this problem, you will come up with a more efficient protocol.

   (a) Assume that Bob already has a set $\{X_1, X_2, \ldots, X_l\}$ of at most $(n+1)/2$ messages. Each message is of length m and exactly one of them is the correct message X. The

[1] Note that even if Bob did not know m, the good Alices could ensure he learns it by sending a small number of bits

good Alices want to tell Bob which is the correct message while sending only a fairly small number of bits. They use fingerprints in the following way. Each good Alice chooses a random prime p and sends the numbers X mod p and p to Bob. (Note that each good Alice is independently choosing her own random prime). Bob then receives a set $\{f_1, f_2, \ldots, f_{l'}\}$ of fingerprints (some from the good Alices and some from the bad Alices). He checks each fingerprint against each of the messages and accepts only that message which matches a majority of the fingerprints.

Question: If each good Alice chooses her prime in the range 1 to τ, what should τ be to ensure that this scheme fails with probability no more than $O(1/n)$. What then is the total number of bits that the good Alices must send? Hint: Argue that for this scheme to work, it suffices to ensure that no fingerprint sent by a good Alice matches a single incorrect message. Then choose τ so that this event occurs with low probability.

(b) Now consider the following two round protocol for sending the message X to Bob. In round one, each good Alice with probability p sends the entire message X to Bob; with probability $1 - p$, she sends nothing. In round two, the good Alices use the protocol described in part (a) above.

Question: How large must p be to ensure that Bob receives the message X from at least one good Alice with probability $1 - O(1/n)$ in the first round? What is the expected number of bits sent by good Alices in round one? Can you say that the actual number of bits sent by good Alices is close to this expected value with high probability? What is the total expected number of bits sent by good Alices in the entire protocol and the probability of error for the entire protocol?